

# REALLOC

Not suitable for use with secure memory because memory contents are not zeroed out

Sean Barnum, Cigital, Inc. [vita<sup>1</sup>]

Copyright © 2007 Cigital, Inc.

2007-04-02

## Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 3399 bytes

Attack Category	<ul style="list-style-type: none"><li>Memory Scanning</li></ul>								
Vulnerability Category	<ul style="list-style-type: none"><li>Information Leakage</li></ul>								
Software Context	<ul style="list-style-type: none"><li>Memory Management</li></ul>								
Location									
Description	<p>realloc() is not suitable for use with secure memory because memory contents are not zeroed out.</p> <p>The realloc() function takes an allocated memory block and expands (or contracts) it to a bigger (or smaller) size. This may involve moving the chunk of memory and copying over the old contents. When this is done, the old contents are discarded and left in memory somewhere. For secure memory applications where it is important to erase all traces of data, this behavior is inappropriate.</p> <p>Realloc() has a variety of other sensitive issues related to reliability. Since it moves memory around, any old pointers to that memory become invalid and could cause the program to crash or otherwise misbehave.</p>								
APIs	<table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>realloc</td><td></td></tr></table>			Function Name	Comments	realloc			
Function Name	Comments								
realloc									
Method of Attack	Sensitive data may be left in memory and could potentially be accessed by an attacker.								
Exception Criteria									
Solutions	<table><tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr><tr><td>Sensitive data needs to be in a resized memory block.</td><td>Do not use realloc for secure memory. If no secure version of realloc is available,</td><td>Effective. Reliability risks still possible; however, if old copies of pointer are left around.</td></tr></table>			Solution Applicability	Solution Description	Solution Efficacy	Sensitive data needs to be in a resized memory block.	Do not use realloc for secure memory. If no secure version of realloc is available,	Effective. Reliability risks still possible; however, if old copies of pointer are left around.
Solution Applicability	Solution Description	Solution Efficacy							
Sensitive data needs to be in a resized memory block.	Do not use realloc for secure memory. If no secure version of realloc is available,	Effective. Reliability risks still possible; however, if old copies of pointer are left around.							

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

	use malloc and a secure version of memset(). See MEMSET rule for appropriate usage.				
<b>Signature Details</b>	void *realloc(void *ptr, size_t size)				
<b>Examples of Incorrect Code</b>	<pre>[...] ptr = realloc(ptr, NEW_SIZE); [...]</pre>				
<b>Examples of Corrected Code</b>	<pre>[...] char * newptr = malloc(NEW_SIZE); memset(newptr, 0, NEW_SIZE); memcpy(newptr, ptr, min(OLD_SIZE, NEW_SIZE)); secureMemset(ptr, 0, OLD_SIZE); /* doesn't get optimized away */ ptr = newptr; [...]</pre>				
<b>Source Reference</b>	<ul style="list-style-type: none"> <li>man page for realloc()</li> </ul>				
<b>Recommended Resource</b>					
<b>Discriminant Set</b>	<table> <tr> <td><b>Operating Systems</b></td><td> <ul style="list-style-type: none"> <li>Windows</li> <li>UNIX</li> </ul> </td></tr> <tr> <td><b>Languages</b></td><td> <ul style="list-style-type: none"> <li>C</li> <li>C++</li> </ul> </td></tr> </table>	<b>Operating Systems</b>	<ul style="list-style-type: none"> <li>Windows</li> <li>UNIX</li> </ul>	<b>Languages</b>	<ul style="list-style-type: none"> <li>C</li> <li>C++</li> </ul>
<b>Operating Systems</b>	<ul style="list-style-type: none"> <li>Windows</li> <li>UNIX</li> </ul>				
<b>Languages</b>	<ul style="list-style-type: none"> <li>C</li> <li>C++</li> </ul>				

## Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at [copyright@cigital.com](mailto:copyright@cigital.com)<sup>1</sup>.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>